

Research Article

Stationary Hand Gesture Authentication Using Edit Distance on Finger Pointing Direction Interval

Alex Ming Hui Wong¹ and Dae-Ki Kang²

¹Department of Ubiquitous IT, Graduate School, Dongseo University, 47 Jurye-Ro, Sasang-Gu, Busan 47011, Republic of Korea

²Department of Computer & Information Engineering, Dongseo University, 47 Jurye-Ro, Sasang-Gu, Busan 47011, Republic of Korea

Correspondence should be addressed to Dae-Ki Kang; dkkang@dongseo.ac.kr

Received 12 July 2016; Revised 28 September 2016; Accepted 17 October 2016

Academic Editor: HuaPing Liu

Copyright © 2016 A. M. H. Wong and D.-K. Kang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

One of the latest authentication methods is by discerning human gestures. Previous research has shown that different people can develop distinct gesture behaviours even when executing the same gesture. Hand gesture is one of the most commonly used gestures in both communication and authentication research since it requires less room to perform as compared to other bodily gestures. There are different types of hand gesture and they have been researched by many researchers, but stationary hand gesture has yet to be thoroughly explored. There are a number of disadvantages and flaws in general hand gesture authentication such as reliability, usability, and computational cost. Although stationary hand gesture is not able to solve all these problems, it still provides more benefits and advantages over other hand gesture authentication methods, such as making gesture into a motion flow instead of trivial image capturing, and requires less room to perform, less vision cue needed during performance, and so forth. In this paper, we introduce stationary hand gesture authentication by implementing edit distance on finger pointing direction interval (ED-FPDI) from hand gesture to model behaviour-based authentication system. The accuracy rate of the proposed ED-FPDI shows promising results.

1. Introduction

Hand gesture recognition has been adapted into some of our daily-use home appliances [1, 2], in electronic devices [3–5], and even in vehicles [6] as a method of input and it is said to be the interface of Internet-of-Things (IoT) in the future [7]. But with such a wide usage and implementation, security issues are bound to emerge. Due to these issues, hand gesture passwords have been researched since the past few years [8–10]. But hand gesture password is not sufficiently secure as hand gesture can be easily learned and imitated. In order to overcome this limitation, we have adopted behaviour into the hand gesture, in return enhancing hand gesture password into hand gesture authentication.

It is interesting that hand gesture recognition has been researched throughout the decades, but there has not been any clear definition or classification on it. In this paper, we classified hand gesture into two major categories: static hand gesture recognition and motion hand gesture recognition.

Static hand gesture recognition is performed with captured images; it can be using only one image or more than one. Motion hand gesture recognition includes all the movement between different hand gestures; the recognition process is conducted over a flow of hand gesture from the beginning until the end of the gesture. That is, static hand gesture is analogous to still image, whereas motion hand gesture is to video. In motion hand gesture recognition, there are other subcategories such as dynamic hand gesture recognition, stationary hand gesture recognition, finger gesture recognition [1, 11], and arm gesture recognition [12–14]. Detailed information on hand gesture recognition is explained in Section 3.1.

User authentication can be implemented through various methods including biometric feature based ones [15]. There are two main approaches in biometric authentication: physiology-based and behaviour-based. Physiology-based approach includes fingerprint, iris, and voice; whereas behaviour-based or behaviorometrics [16] includes human

gesture, keystroke dynamic on keyboard, mouse, and touch screen [17]. Based on [18], different people develop different behaviour even when performing the same gesture or movement. These behaviours can be evaluated through movement speed, acceleration, positioning, distance travelled, and so forth.

There have been a considerable amount of research on biometrics authentication and some of them are based on hand gesture. Keystroke dynamics [17], mouse biometrics [16], multitouch gesture-based authentication [19], digital signature with accelerometer (e.g., uWave), and other approaches [11] have shown promising results in authentication, but they still require users to have contact with the devices. Some hand gesture authentication [8] is not practical in real-life application as it only uses a single capture image of the hand gesture and can also be easily mimicked by imposters, whereas the others [12, 13] require more room to perform and have very few available gestures that can be done as compared to stationary hand gesture authentication. Stationary hand gesture authentication does not require user to come into contact with the device, nor is it a trivial image based authentication, and it does not require a lot of room to be performed. In addition, it is more reliable, user-friendly, and secure compared to other hand gesture authentications which will be discussed in detail in Section 3.1.2.

In this paper, we have applied edit distance algorithm to stationary hand gesture behaviour to authenticate user by comparing the dissimilarities between the finger's states based on the time interval. Mean and standard deviation method and acceptance rate method are set as the thresholds to accept or reject the data. As for the result evaluation and analysis, accuracy from the confusion matrix and equal error rate (EER) from receiver operating characteristic (ROC) curve have been considered. The detailed methodology is explained in Section 4.

The outline of this paper is as follows: in Section 2, we explain the related work. The following section, Section 3, discusses hand gesture, Leap Motion controller, and edit distance algorithm. Section 4 explains our proposed method ED-FPDI, followed by results and discussion in Section 5. Finally, in Section 6, we close the paper with conclusion and present our future work.

2. Related Work

Chan et al. [20] have used Leap Motion controller for authentication via hand geometry and gestures. They have implemented two types of authentication in their experiment which is the static authentication and continuous authentication. In static authentication scenario, users have been asked to draw a circle with one finger; in continuous authentication scenario, users have been asked to perform basic actions, such as key or screen tapping, scrolling, swiping, and other actions. These data are being evaluated using different features: hand and finger properties, radius of the circle drawn, acceleration of hand, pinching and grabbing strengths, and so forth. These features are then classified using random forest classifiers and the result has shown that more weights are given to the physical properties of the hand and finger. The accuracy of the

experiment is 99.97% on the static authentication and 98.39% on the continuous authentication.

Fong et al. [8] have proposed a biometric authentication model using static hand gesture images. Participants have been asked to perform American Sign Language of the 26 letters and each sign language was captured using a RGB camera. Features of the hand and finger, such as the position and angle, are extracted from the image for classification. The authors have used ten machine learning algorithms which are chosen from the major types of classification, including decision trees, rule-based methods, kernel functions, and Bayes methods. The experiment has shown promising overall result with a maximum accuracy of 93.75%.

Liu et al. [11] have proposed uWave, an efficient recognition algorithm for a single three-axis accelerometer such as the Nintendo Wii Remote. This method is similar to that of a signature-based recognition where users use a handheld device and start scribbling onto a flat screen to form a pattern. The pattern is then compared with other patterns using dynamic time warping (DTW) to authenticate the users.

Lai et al. [12] have proposed a user identification method using Microsoft Kinect to detect the arm gesture of the user. The arm gesture has been recorded in the form of body silhouette and is compared with other recorded gestures using nearest neighbor (NN) algorithm to identify user. Wu et al. [13] have expanded the approach of Lai et al. [12] by adopting Kinect skeleton model instead of body silhouette and extending the method to perform user authentication instead of only identification. The DTW algorithm has been used to compare the gestures. Unlike uWave, these approaches do not require user to equip or hold any devices on them.

Sayed et al. [16] have proposed biometric authentication using mouse gesture to create pattern by moving the mouse around. The method which attempts to find the time differences between two similar patterns is learning vector quantization (LVQ) neural network.

On the other hand, Sae-Bae et al. [19] have proposed authentication on multitouch devices. By comparing the touch sequence and time interval of the sequence using DTW algorithm, they have been able to authenticate users through touchscreen devices. The authentication is not limited to only one touch sequence.

Both Chan et al. [20] and Fong et al. [8] are closely related to our proposed method. Similar to Fong et al., we have used hand gestures identical to that of sign languages, but, instead of using static images, we have recorded these gestures in motion which includes the gestures and also the changes between each gesture. Chan et al. [20], on the other hand, have implemented finger gesture authentication through Leap Motion controller with gestures such as drawing, scrolling, and tapping. But their results show that most of the weights for authentication are placed into the physical properties of the hand and finger instead of the gesture. In comparison, Fong et al.'s method [8] can be easily copied by other users since it is just static hand gesture, whereas Chan et al. method [20] focuses more on hand properties which can easily be done using hand sculpture for authentication. Fong et al.'s method [8] is undeniably accurate in authentication,

but, in some situations, it may be easy for imposters to copy static hand gesture because the method considers simple static images, whereas Chan et al.'s method [20] focuses more on hand properties which are considered as physiological biometrics instead of behavioural based biometrics, and it may also be insecure at some point as imposters can trace the genuine hand and duplicate it.

Recently, Liu et al. [21, 22] used dynamic time warping (DTW) [21] and canonical time warping (CTW) [22] to develop the kernel sparse coding method to analyze the time series to improve the performance of object recognition.

To our knowledge, we have yet to come across similar approaches to our proposed method which uses stationary hand gesture in authentication, as most of the research has been based on static hand gesture, arm gesture, and finger gesture. A comparison on these different hand gestures and the advantages of stationary hand gesture over other hand gesture is stated in Sections 3.1.1 and 3.1.2. As for other related work, due to the difference in methodology used, we have constructed a table of comparison in Table 8 and discussed in Section 5.3.2 as to why gesture-based authentication is more preferable. Since there are also different types of gesture recognition available, a comparison among different gesture recognition has been constructed in Table 9.

3. Background

3.1. Hand Gesture and Its Behaviour

3.1.1. Categories of Hand Gesture. Although keyboard and mouse are the traditional way of providing input to the computer, they are considered unnatural for human interaction [23, 24]. Hand gesture as an input has advantages over traditional input methods, such as being handsfree, able to communicate with the device from a distance, and more natural due to daily usage of hand gesture in human-to-human (H2H) communication[25, 26].

Hand gesture is a very vague term as there are so many different types of hand gesture available. Therefore, we have categorized hand gesture in Figure 1.

The terms used in hand gesture classification are as follows:

- (i) Static means no movement or motion on the subject at all, usually in single image form.
- (ii) Motion means there is movement involved.
- (iii) Stationary means no movement on the particular subject, but objects that do not affect the movement of the subject can still be moving; for example, stationary hand gesture is when the hand, wrist, and arm (since movement of the wrist and arm will affect the position of the hand) are not moving or having any motion, whereas the fingers are free to move as long as this does not affect the position of the hand.
- (iv) Dynamic means any part of the subject can have movement of motion even if it affects other objects; for example, dynamic hand gesture allows hand to

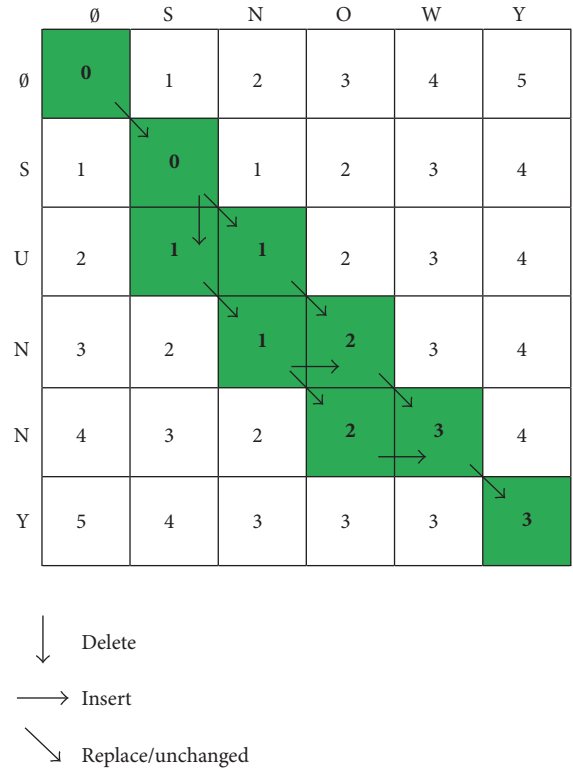


FIGURE 1: Edit distance matrix of string "SUNNY" to "SNOWY."

move freely and, at the same time, fingers' position will be affected.

In some papers [12, 13], they refer to arm gesture as hand gesture which cause a lot of confusion; thus, we separate them into two different categories (Table 1). As for hand gesture, there are two major types, which are static hand gesture and motion hand gesture. We have defined two types of motion hand gesture: stationary hand gesture (hand and wrist are not moving but fingers are moving) and dynamic hand gesture (hand, fingers, and wrist are moving). When both arm and hand are considered into the recognition, we refer to it as arm with hand gesture. Lastly, as for drawing, scribbling, and swiping with just the finger, we refer to them as finger gesture. There are two subcategories in finger gesture, which are hand finger gesture (finger movement is only affected by the movement of the hand and fingers) and arm finger gesture (finger movement is affected by the movement of the arm, hand, and fingers). In our experiment, we focus on only stationary hand gesture-based authentication.

3.1.2. Advantages of Stationary Hand Gesture over Other Hand Gesture Approaches. There have been a few researches and publications on hand gesture authentication and its behaviour such as using hand gesture image (static hand gesture), dynamic hand gesture, arm gesture, and finger gesture (draw shape or signature using fingertip). But until now, there has yet to be any research on stationary hand gesture authentication. Moreover, there are reasons to why we choose stationary hand gesture over the other approaches:

TABLE 1: Category of hand gestures.

Gesture	Type
Arm gesture	Static arm gesture
	Motion/dynamic arm gesture
Hand gesture	Static hand gesture
	Motion hand gesture
	Dynamic hand gesture (motion on hands, fingers, and wrist)
	Stationary hand gesture (motion on fingers only)
Arm with hand gesture	Static arm with hand gesture
	Motion/dynamic arm with hand gesture
Finger gesture	Hand finger gesture (drawing or swiping with hand and finger only)
	Arm finger gesture (drawing or swiping with the arm and finger)

- (i) Hand gesture image can be easily mimicked. It authenticates user by detecting the position, angle, and physical properties of hand on just one image. This image could probably be printed out and used on the system by other users or imposters.
- (ii) Dynamic hand gesture will probably be the best in terms of authentication out of all the other hand gesture authentication methods, but it is difficult to compute since there are too many factors that need to be examined such as the gestures of finger, hand, palm, wrist, and arm. In addition, there has yet to be any reasonable device in accurately recognizing the entire arm, hand, and fingers together.
- (iii) Arm gesture requires a larger space to be performed. It is also very difficult to conceal the gesture from outsiders. In addition, there are not many arm gestures available besides swinging up and down, left and right.
- (iv) Finger gesture can be effective on simple gesture but sometimes can also be easily copied by others. Complex gesture, on the other hand, can be more secure but user may need visual cue on their gesture to prevent drawing in the wrong order or mispositioning.

Stationary hand gesture cannot be easily mimicked due to different behaviour; it does not need complicated computational algorithm or large space to perform, and it does not necessarily need any visual support while performing.

3.1.3. Stationary Hand Gesture Behaviour. Even though hand gesture can be used as password, but the issue with hand gesture is that it can be easily mimicked; however, due to the development through habits, behaviour in the gesture cannot be easily copied [27]. For example, when a hand is moving from an open to a close gesture (stretching out all fingers to retracting all fingers into the palm), different people will have distinct timing, moving angle, and position while doing the same gesture. It may not be impossible to mimic one's behaviour, but it surely takes a huge amount of effort which may take years to perfect it. Moreover, behaviour may change

over time depending on gender, age, environment, and so forth. Therefore, before perfecting a particular person's behaviour, he/she may have changed or developed new behaviour.

Stationary hand gesture behaviour can be determined by, but not limited to, the finger's moving speed, acceleration, position, and moving angle. In our experiment, we have adapted finger pointing direction as the feature of hand gesture behaviour. It features the pointing direction of each finger based on the finger's state, which are open (stretch out) and close (retract) time interval. The time interval can be either long or short depending on various users. Leap Motion controller has been used to record all the data in our experiment. The data contain each finger's information including the time intervals and pointing directions. These behaviours can be easily distinguished by computer but not human, since the time interval is calculated in microseconds.

3.2. Leap Motion Controller. Leap Motion controller is a small sensor device that tracks hand, finger, or pointed object, either in static state or in motion state as input with very high accuracy and precision [28, 29]. It outperforms many other sensor devices, in terms of hand gesture detection or recognition, such as Microsoft Kinect and Creative Senz3D [30]. Moreover, the Leap Motion controller can record hand gesture at the rate of over 200 frames per second (fps) [31], but, in our experiment, the recorded data is roughly around 110 fps only. Nevertheless, it is sufficient to determine the behaviour of the gesture as the interval between each frame is denoted in microseconds.

3.3. Edit Distance. Edit distance [32] or Levenshtein distance is used to find the minimum number of operations required between two strings (or words). There are three types of operations in edit distance, insert, delete, and replace, where each transformation costs one operation. An example is to change the string "SUNNY" to "SNOWY": "S" remains, delete "U", first "N" remains, replace second "N" with "O", insert "W", and "Y" remains. This string change costs three

operations which includes delete, replace, and insert. The algorithm for edit distance is shown in

$$E(i, j) = \min \begin{cases} 1 + E(i - 1, j), & \text{delete} \\ 1 + E(i, j - 1), & \text{insert} \\ \text{diff}(i, j) + E(i - 1, j - 1), & \text{replace,} \end{cases} \quad (1)$$

where i is the length of the first string and j is the length of the second string.

Figure 1 shows the edit distance matrix where each operation costs one point. As shown in the figure, going down the matrix is the operation delete, going right is the operation insert, and going diagonally down right is the operation replace or remain unchanged. The very end of the matrix, which is located at the lower right corner of the matrix, is considered as the shortest path or minimum number of operations needed for the transformation of two strings.

In authentication, accuracy and speed are the most important aspects. We have chosen edit distance in experiment due to its speed, simplicity, and efficiency in detecting the differences in data. Edit distance saves both computational and authentication time and gives accurate prediction on either genuine user or imposter.

4. ED-FPDI

In this paper, we introduce a novel approach for stationary hand gesture authentication, edit distance on finger pointing direction interval (ED-FPDI). Our experiment consists of five phases: recording phase, time interval normalization phase, data filtration phase (starting and ending point filtration), training and testing phase, and result evaluation phase. Time interval normalization phase, data filtration phase, and training and testing phase are also the core phases of ED-FPDI. The overall procedure of the experiment is illustrated in Figure 2.

4.1. Recording Phase. Vatavu and Zaiti [1] have recorded and published Leap Motion hand gesture dataset for remote control of devices. Their dataset is very useful for remote control experiments, but it does not contain stationary hand gesture data except open palm and close palm. Other gestures recorded in their dataset are mostly swinging hands in different direction and drawing letters or shapes (finger gesture). Although there were eighteen different participants data recorded, the data recorded is not used for authentication, but it is used as a remote control for television functions, such as changing channel and volume, opening menu, and showing TV guide. Therefore, their dataset is not directly suitable for our proposed stationary hand gesture authentication.

Thus, to our knowledge, there has yet to be any stationary hand gesture dataset recorded using Leap Motion controller available; thus we have to conduct our own recording data. The setup for the recording session of the experiment is demonstrated in Figure 3.

There are a total of ten participants, including seven males and three females, in their 20s, who participated in

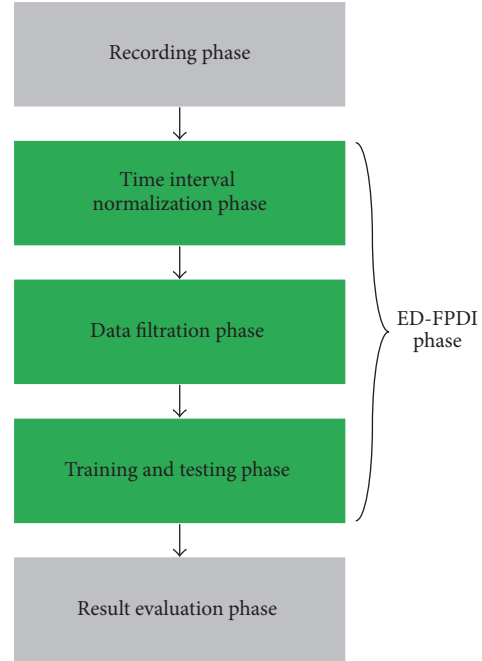


FIGURE 2: Flowchart of the experiment procedure.

our experiment. All gestures are done with only the right hand. We have asked the participants to perform the same two sets of hand gestures at their own pace while keeping them as consistent as possible. Nine of them were asked to perform 25 times on a hand gesture which are used as the test data of the experiment, also known as the imposters, whereas one of the participants, who serves as the control group which in this case known as the genuine user, was asked to perform 125 times on the hand gesture, where 100 instances of data are used as the training data while the remaining 25 instances of data are used as the test data. In summary, a total of 350 instances of data on a single gesture were recorded; 100 instances of data from the genuine user are used for training, while 250 remaining instances of data are used for testing, where 25 instances of data are from the genuine user and the remaining 225 instances of data are from the imposters. Training data and test data from the genuine user are independent. The hand gestures recorded are known as “205” and “7631” which are shown in Figures 4 and 5, respectively. It is worth mentioning that these gestures are defined by the genuine user and are not restricted to these gestures only, and the experiment makes the assumption that all the imposters have already known the hand gestures or password gestures [12] (since the hand gestures are considered as passwords in our experiment) that the genuine user has inputted.

Figure 6 shows the axes of the Leap Motion controller. The hand will be placed on top of the controller, parallel to z -axis of the controller. In our experiment, the hand direction is always pointing towards the negative z -direction (pointing towards the monitor as shown in Figure 6) between -1.0 and -0.9 in unit vector, whereas x -direction and y -direction of

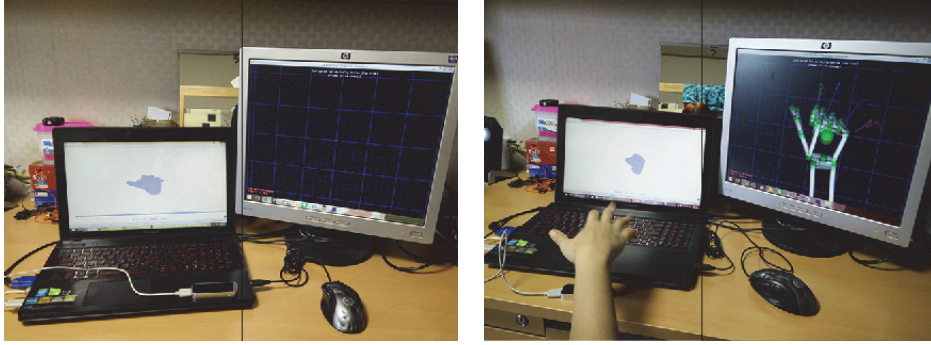


FIGURE 3: Setup of the hand gesture recording. Left monitor is the recorder, whereas the right monitor is the top view of the hand. The top view is necessary to detect errors and inaccuracies of hand gestures during recording.



FIGURE 4: Hand gesture for “205.”

the hand are as closed to 0 in unit vector as possible (even when performing gesture).

While recording the gestures, we have found out that gesture “7631” produces a lot of errors from the Leap Motion controller while transforming from “6” to “3” and from “3” to “1.” The errors have occurred for more than half of the time during recording for each user, meaning that each user has been recording for more than 50 times just to arrive at 25 instances of data without any error. It has also worthwhile to note that the speed of each user on this gesture varies by a big margin including the genuine user and therefore it may influence the accuracy of the experiment. As mentioned before, these errors were not accidentally or purposely made by the user but by the detection inaccuracy of the Leap Motion controller or software as shown in Figure 7.

We would like to note that the hand gestures in our experiment are not static; they are motion gesture which changes from one static gesture to another static gesture. For example, it changes from “2” fingers to “0” fingers and the motion between the two gestures is considered in our experiment.

4.2. Time Interval Normalization Phase. This phase is the combination of normalizing time interval and filtering data steps. From the recorded data, the intervals between each frame of the data are nonconsistent and therefore have to be

normalized before comparison. We have adopted linear interpolation to normalize the interval. For example, the intervals can be 0, 12345, 23456, and 34567 microseconds in the recorded data. After normalization, they will be 0, 10000, 20000, and 30000 microseconds. This has to be done in order for every data to have a consistent interval for ease of comparison when using edit distance algorithm.

4.3. Data Filtration Phase. Our proposed ED-FPDI considers only two timing elements from each finger which are open finger and close finger. More detailed description of the two timing elements is shown as follows:

- (1) Open finger (O): stretching the finger outwards (when the pointing direction along z -axis is between -0.8 and -1.0 in vector unit)
- (2) Close finger (C): retracting the finger into the palm (when the pointing direction along z -axis is larger than -0.8)

The reason that these threshold numbers are chosen as the timing elements is based on the observation that most of open finger data (“O”) in the recorded dataset lie on positions between -0.8 and -1.0 in z -axis. We have applied only two segmentations due to the inaccuracy of the Leap Motion in detecting when the fingers are fully closed as shown in

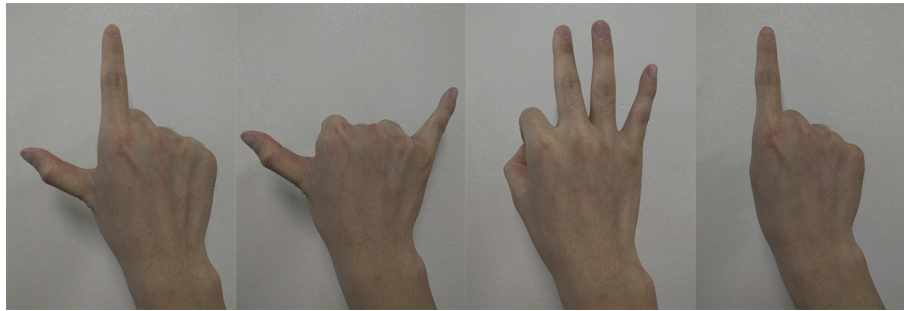


FIGURE 5: Hand gesture for “7631.”

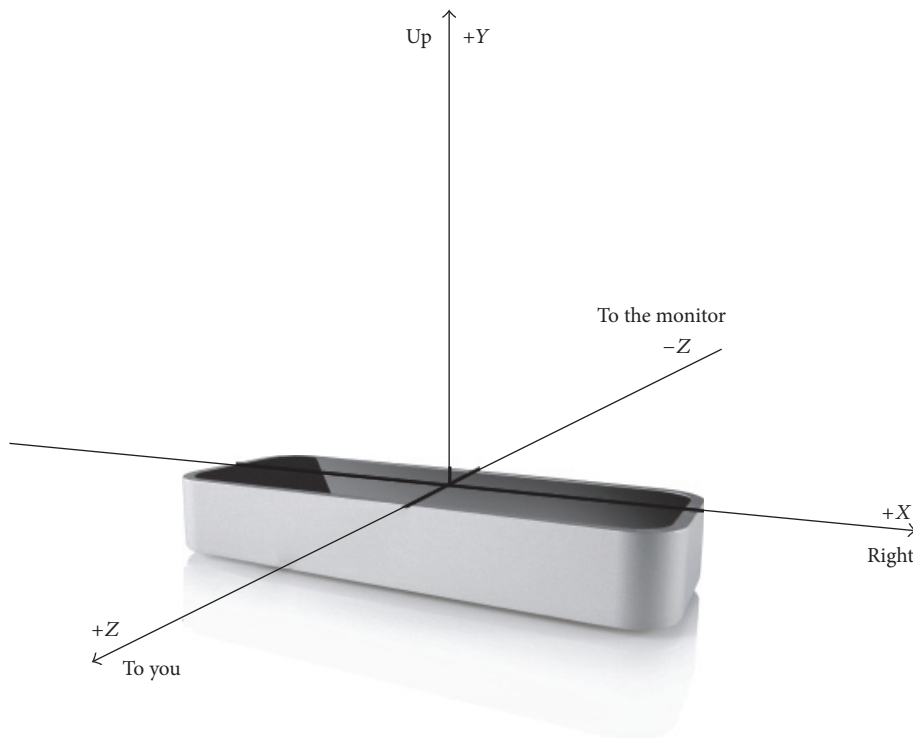


FIGURE 6: Leap Motion controller and its axes.

Figure 7. It can be seen that even when the fingers are completely closed, the Leap Motion controller mistakenly reports that the fingers are only halfway closed. We have observed that the motion from open to close (and vice versa) is supposed to be around -0.8 to 0.8 , while the close element (“C”) is between 0.8 and 1.0 in unit vector. However, again due to the errors from the Leap Motion controller, these measurements are unable to be incorporated into our system. Note that this observation is considered under the assumption when the hand pointing at z -direction is between -0.9 and -1.0 in unit vector (pointing forward with the Leap Motion controller), and the pointing y -direction is around 0 in unit vector (parallel with the Leap Motion controller). Throughout the experiment, we have also found out that only three fingers (the index, the middle, and the ring fingers) have exhibited consistent precision of pointing direction along z -axis, between -0.8 and -1.0 in vector unit. Due to the large margin

differences in the pointing direction of the thumb and the little fingers varying from different people, they are not taken account in our experiment; however, the experimental results do not show significant degradation. We show two samples in Table 2 to investigate these observations.

From Table 2, we can see that Data 1 forms a string of “OCCCCCCCCOO,” while Data 2 forms “OOCCCCCO.” After applying edit distance algorithm from Data 1 to Data 2, we obtain the minimal edit distance, $E(i, j) = 4$, which means that there is a minimum of four operations to change from Data 1 to Data 2.

But before the application of edit distance to the data, we need to filter out the excess starting gesture and ending gesture. Detailed visualization of the raw data and the filtered data is depicted in Figure 8. As shown in Figure 8(a), the raw data have not defined the starting and the ending point of the gesture. Therefore, we filter out the excess gesture before

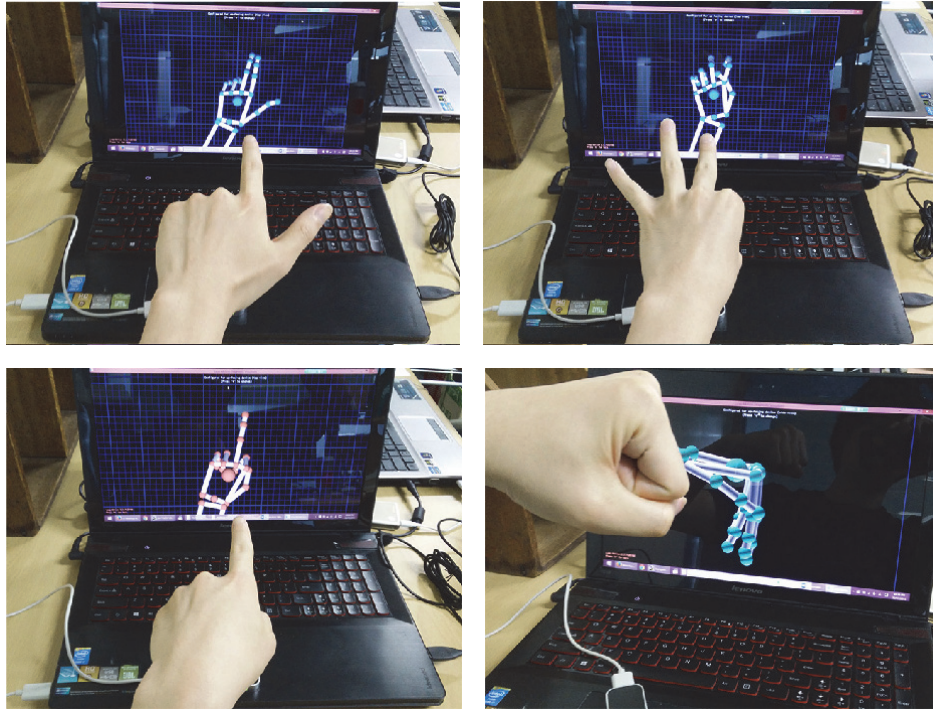


FIGURE 7: Hand gesture errors and inaccuracies shown by the Leap Motion controller while performing hand gestures.

TABLE 2: Examples of time interval for a finger from two different instances of data. Value lesser or equal to -0.8 is considered open (O), while value larger than -0.8 is considered close (C).

Interval (ms)	10	20	30	40	50	60	70	80	90	100	110
Data 1	-0.9	-0.8	-0.5	-0.1	0.4	0.8	0.3	-0.2	-0.7	-0.9	-0.9
Data 2	-0.8	-0.9	-0.8	-0.2	0.1	0.4	-0.4	-0.7	-0.8	—	—

the starting and after the ending point illustrated in Figure 8(b), by detecting the initial movement of the finger from a predefined starting hand gesture (usually hand gesture of “5”). Similarly, we detect the ending point by locating the point where the hand gesture changes from the last gesture to a predefined ending hand gesture (usually hand gesture of “5”). If the starting point or ending point of the gesture is actually “5” (such as the case “205”), then another predefined hand gesture is used. Note that this filtering detection is done manually. The purpose of filtering is to ensure the consistent starting point and the ending point for every gesture.

4.4. Training and Testing Phase

4.4.1. Mean and Standard Deviation (Me&SD) Method. After the completion of filtering and segmenting the raw data, we apply edit distance algorithm to the processed dataset. First of all, we calculate pairwise edit distances $E(\text{Tr}_i, \text{Tr}_j)$ for the 100 training instances, where E is the edit distance and Tr is the training data. The total number of edit distance calculations for each pair of instances in the training data is 4,950, which can be easily obtained from a combination formula of $\binom{100}{2}$. From the calculated edit distance values, we find the mean and standard deviation to acquire the threshold interval. Note

that we have three records (or fixed-length sequences) in one training instance of which each record corresponds to one finger. In other words, we have three threshold intervals (i.e., each of them for one finger) in one training instance. The acquired threshold intervals are used as guidelines for authentication.

After the training phase, we apply the 250 test instances for edit distance calculation with the 100 training instances. Each test instance is compared with the 100 instances of training data and return 100×3 values of $E(\text{Te}, \text{Tr})$, where E is the edit distance and Te is the test data. Note that we multiply by 3 because we calculate the mean of 100 $E(\text{Te}, \text{Tr})$ values for one finger and we repeat the same procedure for other two fingers. For each finger in the test instance, we compare its mean with the corresponding threshold interval estimated from the training data. If the mean is within the threshold interval, we mark that particular finger in the test instance as accepted. And if all three fingers in the test instance are accepted, we consider the user as a genuine user, otherwise as an imposter. One example of these procedures is shown in Table 3.

From Table 3, Data 1 will be considered as a genuine user because all of its mean values of each finger are within the threshold interval. Data 2 and Data 3 will be noted as imposters since at least one of the mean values of each finger in

TABLE 3: Examples of threshold intervals from the training data using the mean and standard deviation and mean distances (differences) from each test instance with training instances.

Finger	Index	Middle	Ring
Threshold	14.24~43.12	14.98~42.26	11.81~38.24
Data 1	25	26	24
Data 2	10	72	68
Data 3	50	41	37

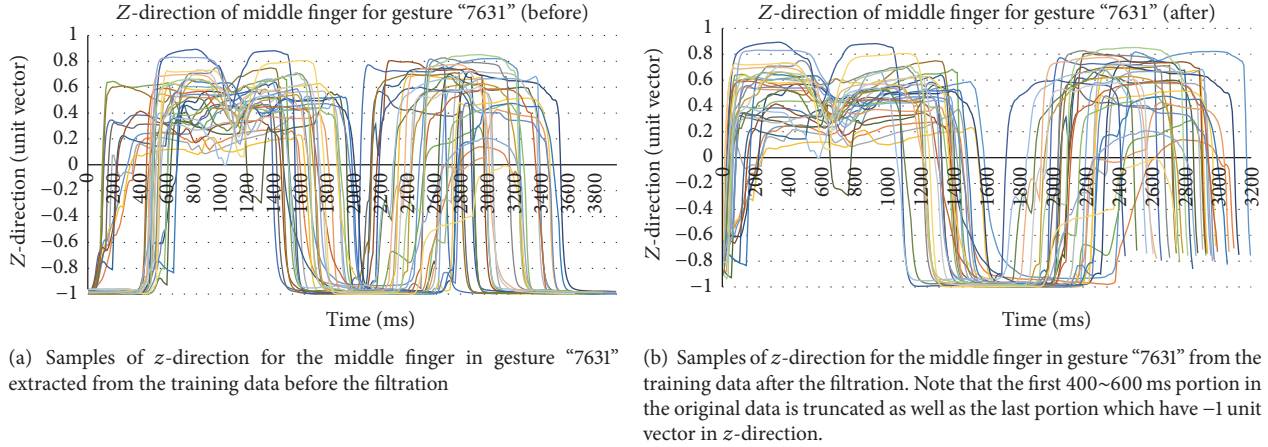


FIGURE 8: Z-direction of middle finger for raw data and filtered data.

the data is outside of the threshold interval. Pseudocode 1 demonstrates the pseudocode of our mean and standard deviation (M&SD) method, where M is the mean and SD is the standard deviation. It can be seen that the asymptotic upper bound of M&SD method's training time lies in $O(n^2m^2)$ where $n = |T_r|$ is the number of training instances and $m = \max_i |T_{r_i}|$ is the length of the longest training instance. The asymptotic upper bound of M&SD method's test time is $O(nNm^2)$ where $n = |T_r|$ is the number of training instances, $N = |T_e|$ is the number of test instances, and $m = \max_i |T_{r_i}|$ is the length of the longest training instance.

4.4.2. Acceptance Rate (AR) Method. In addition to mean and standard deviation (M&SD) method, we have discussed in Section 4.4.1 that we propose acceptance rate (AR) method, which uses the acceptance rate as a threshold calculated from the training data. For calculating the acceptance rate, we apply mean and standard deviation obtained from the training data to the training data itself. The main idea is that, for example, if the acceptance rate of the training data is 0.5, then the acceptance rate of test data should be close to 0.5 to be accepted as that of a genuine user. In other words, if the acceptance rate of the test data is far from the acceptance rate of the training data, the chance of being accepted as that of a genuine user is low.

In order to calculate the acceptance rate of the training data (AR_{Tr}), our algorithm compares the M&SD threshold found from the previous method with all the other 4950 edit distances $E(Tr_i, Tr_j)$ in the training data. The M&SD threshold from the training data is then applied to the test data to

obtain the 100 edit distances $E(Te, Tr)$. In order to obtain the acceptance rate of the test data (AR_{Te}), the 100 $E(Te, Tr)$ are compared with the same M&SD threshold from the training data. Then our algorithm uses the acceptance rate of training data (AR_{Tr}) to compare with the acceptance rate of the test data (AR_{Te}). The difference of AR_{Tr} and AR_{Te} is used to form the ROC curve. Note that the acceptance rate can be estimated in two ways: the average acceptance rate of all fingers (Average AR) and the acceptance rate of each finger (Each AR). The pseudocode for this approach is shown in Pseudocode 2.

The asymptotic upper bound of AR method's training time is $O(n^2m^2)$, where $n = |T_r|$ is the number of training instances and $m = \max_i |T_{r_i}|$ is the length of the longest training instance. And the asymptotic upper bound of AR method's test time is $O(nNm^2)$, where $n = |T_r|$ is the number of training instances, $N = |T_e|$ is the number of test instances, and $m = \max_i |T_{r_i}|$ is the length of the longest training instance. In terms of asymptotic time complexity, we can see that AR method is as fast as M&SD method.

4.5. Result Evaluation Phase. As mentioned before, we consider two evaluation methods: accuracy from confusion matrix method and EER from ROC curve method. For clear explanation of confusion matrix, we describe a few basic terminologies here:

- (1) True acceptance rate (TAR): genuine user being accepted (good)
- (2) True rejection rate (TRR): imposters being rejected (good)

Mean and Standard Deviation:

Input: Training dataset and test dataset

Output: Hand gesture accepted (genuine user) or rejected (imposter), and ROC curve

Training phase:

begin

- (1) **for** each Tr_i compares with Tr_j , where $i \neq j$
 calculate edit distance $E(Tr_i, Tr_j)$
- (2) calculate M and SD of $\forall E(Tr_i, Tr_j)$
 set $M \pm SD$ as threshold interval

end.

Testing phase:

Begin

- (1) **for** each Te compares with $\forall Tr$
 calculate edit distance $E(Te, Tr)$
- (2) calculate M of $E(Te, Tr)$ in each dataset
- (3) **if** $E(Te, Tr)_M$ of all fingers in a dataset are within threshold interval
 accepted (genuine user)
 else
 rejected (imposter)
- (4) plot ROC curve
- (5) find EER from the ROC curve graph

end.

PSEUDOCODE 1: Pseudocode of the mean and standard deviation (M&SD) method.

Acceptance Rate:

Input: Training dataset and test dataset

Output: ROC curve

Training phase:

begin

- (1) **for** each Tr_i compares with Tr_j , where $i \neq j$
 calculate edit distances $E(Tr_i, Tr_j)$
- (2) calculate M and SD of $\forall E(Tr_i, Tr_j)$
 set $M \pm SD$ as threshold
- (3) **if** $E(Tr_i, Tr_j)$ is within threshold
 $E(Tr_i, Tr_j)_{Accepted}$
 else
 $E(Tr_i, Tr_j)_{Rejected}$
- (4) calculate $AR_{Te} = \text{Total no. of } E(Tr_i, Tr_j)_{Accepted} / \text{Total no. of } E(Tr_i, Tr_j)$

end.

Testing phase:

begin

- (1) **for** each Te compares with $\forall Tr$
 calculate edit distances $E(Te, Tr)$
- (2) **if** $E(Te, Tr)$ is within threshold
 $E(Te, Tr)_{Accepted}$
 else
 $E(Te, Tr)_{Rejected}$
- (3) calculate $AR_{Te} = \text{Total no. of } E(Te, Tr)_{Accepted} / \text{Total no. of } E(Te, Tr)$
- (4) plot ROC curve using difference between AR_{Te} and AR_{Tr}
- (5) find EER from the ROC curve graph

end.

PSEUDOCODE 2: Pseudocode of the acceptance rate method including its evaluation.

TABLE 4: Results on gesture “205.” User number 1 is the genuine user, while the other users are imposters.

User	Accepted	Rejected
1 (genuine)	22	3
2 (imposter)	0	25
3 (imposter)	2	23
4 (imposter)	2	23
5 (imposter)	1	24
6 (imposter)	2	23
7 (imposter)	2	23
8 (imposter)	0	25
9 (imposter)	1	24
10 (imposter)	3	22
Total imposters (225 instances of data)	13	212

(3) False rejection rate (FRR): genuine user being rejected (bad)

(4) False acceptance rate (FAR): imposters being accepted (worse)

For calculating accuracy from confusion matrix method, we calculate it by summing up the true acceptance rate (TAR) with the true rejection rate (TRR) and divide the sum with the total number of instances. However, note that this method is only applicable for mean and standard deviation (M&SD) method in our experiment.

Receiver operating characteristic (ROC) curve [33] has also been used to measure the performance of our algorithm. ROC curve is a graph that illustrates the performance of classifiers by presenting the trade-off between hit rates and false alarm rates while varying the threshold. The advantage of the ROC curve is that we can find the equal error rate (EER) [34] which has been used in most of the biometric security systems to measure the actual performance in imbalanced data. Hence, low EER indicates high accuracy. EER is obtained when both acceptance rate and rejection rate are equal. Both the M&SD method and the acceptance rate (AR) method have been evaluated using this method.

5. Experimental Results and Discussion

Tables 4 and 5 show the experimental results of the proposed algorithm on both the genuine user and imposters about the test gesture “205” and “7631,” respectively. In Table 4, 22 out of 25 instances of genuine user data have been accepted by the system, whereas 13 out of 225 imposters have been accepted. In Table 5, only 20 out of 25 instances of genuine user data have been accepted by the system and 30 out of 225 imposters have been accepted. The numbers of acceptance and rejection for both genuine user and imposters are also shown individually in Tables 4 and 5.

5.1. Accuracy from Confusion Matrix. The equations of true acceptance rate (TAR), false acceptance rate (FAR), and accuracy are explained in (2), (3), and (4), respectively:

$$\text{TAR} = \frac{\text{Genuine user data accepted}}{\text{Total genuine user data}}, \quad (2)$$

TABLE 5: Results on gesture “7631.” User number 1 is the genuine user, while the other users are imposters.

User	Accepted	Rejected
1 (genuine)	20	5
2 (imposter)	0	25
3 (imposter)	0	25
4 (imposter)	4	21
5 (imposter)	3	22
6 (imposter)	9	16
7 (imposter)	0	25
8 (imposter)	7	18
9 (imposter)	2	23
10 (imposter)	5	20
Total imposters (225 instances of data)	30	195

$$\text{FAR} = \frac{\text{Imposter user data accepted}}{\text{Total imposter data}}, \quad (3)$$

$$\text{Accuracy} = \frac{\sum \text{True acceptance} + \sum \text{True rejection}}{\sum \text{Total data}}. \quad (4)$$

From Table 4, TAR, FAR, and accuracy for gesture “205” are calculated as follows:

$$\text{TAR} = \frac{22}{25} = 0.8800,$$

$$\text{FAR} = \frac{13}{225} = 0.0578, \quad (5)$$

$$\text{Accuracy} = \frac{22 + (225 - 13)}{250} = 0.9360.$$

From Table 5, TAR, FAR, and accuracy for gesture “7631” are calculated as follows:

$$\text{TAR} = \frac{20}{25} = 0.8000,$$

$$\text{FAR} = \frac{30}{225} = 0.1333, \quad (6)$$

$$\text{Accuracy} = \frac{20 + (225 - 30)}{250} = 0.8600.$$

From the above calculations, it can be seen that gesture “205” has a higher accuracy of 0.9360, as compared to that of gesture “7631” which is only 0.8600. The reason gesture “7631” has a lower accuracy as compared to gesture “205” has been discussed in Section 4.1 and shown in Figure 7, which is due to the inaccuracy of the Leap Motion controller. Nevertheless, both gestures have shown fairly high accuracy of more than 0.8.

5.2. Equal Error Rate (EER) from Receiver Operating Characteristic (ROC) Curve. ROC curves of gesture “205” and gesture “7631” are depicted in Figures 9 and 10, respectively. The red line represents the results from the mean and standard deviation (M&SD) method, whereas the green line represents the average acceptance rate of all the fingers from

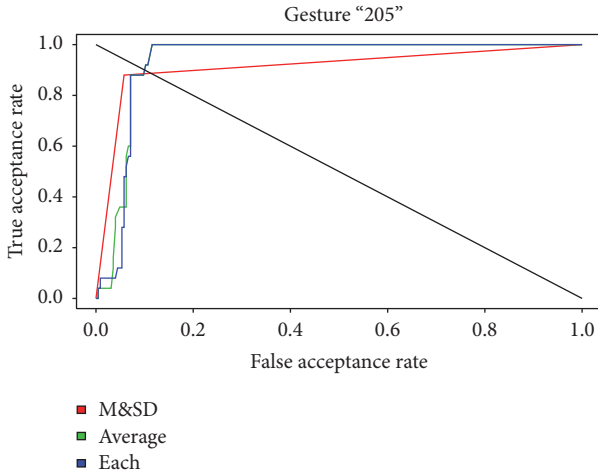


FIGURE 9: ROC curves and the EER line on gesture “205.”

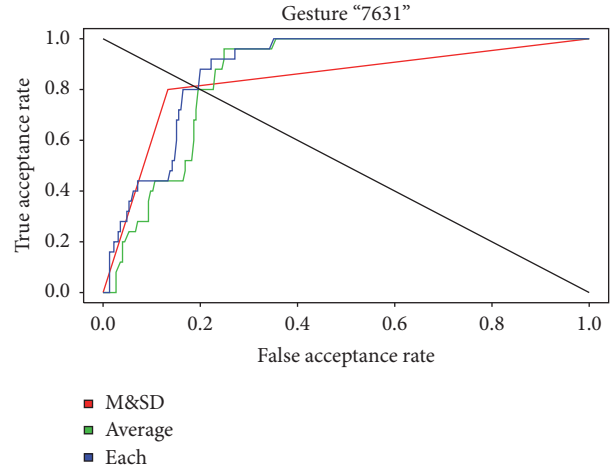


FIGURE 10: ROC curves and the EER line on gesture “7631.”

TABLE 6: The summarized results of both gesture “205” and gesture “7631” using confusion matrix. High TAR, low FAR, and high accuracy indicate high performances.

Gesture	TAR	FAR	Accuracy
205	0.8800	0.0578	0.9360
7631	0.8000	0.1333	0.8600

the training data to be compared with each finger of the test data; and the blue line represents each acceptance rate of the finger from the training data to be compared with the similar finger from the test data. For example, from the training data, the index finger’s acceptance rate will only be compared with the index finger from the test data. The black line that intercepts all those lines through point (0,1) to point (1,0) is the line of EER. The accuracy in ROC curve can be denoted as follows:

$$\text{Accuracy} = 1.0000 - \text{EER}. \tag{7}$$

From ROC curve of gesture “205,” the M&SD method results in EER of 0.1130 and accuracy of 0.8870, whereas both average and each finger acceptance rate method produce EER of 0.1000 and accuracy of 0.9000. As for gesture “7631,” the M&SD method yields EER of 0.1875 and accuracy of 0.8125 and average finger acceptance rate method exhibits EER of 0.2000 and accuracy of 0.8000, while each finger acceptance rate method shows EER of 0.1958 with 0.8042 as accuracy.

5.3. Summarized Results and Discussion

5.3.1. *Summarized Results.* Tables 6 and 7 show the summary of the result evaluations on both gesture “205” and gesture “7631.” Both gestures are evaluated using two different methods, accuracy of confusion matrix, and equal error rate (EER) from ROC curve.

Table 6 is the result in confusion matrix which consists of the following:

- (i) True acceptance rate (TAR) based on (2)

- (ii) False acceptance rate (FAR) based on (3)
- (iii) Accuracy based on (4)

Table 7 is the result in ROC curve which consists of the following:

- (i) Evaluation methods
 - (a) Mean and standard deviation M&SD
 - (b) Average acceptance rate of all fingers from training data compared with test data (Average AR)
 - (c) Acceptance rate of each finger from training data compared with the similar finger from test data (Each AR)
- (ii) Equal error rate (EER) from the interception point of the ROC curve with the straight diagonal line shown in Figures 9 and 10
- (iii) Accuracy based on (7)

In the case of accuracy, higher value indicates higher performance; however, in the case of EER, lower value signifies higher performance.

5.3.2. *Discussion.* As is seen from Tables 6 and 7, there is a difference between the accuracy obtained from the confusion matrix and ROC curve. This is due to the imbalance data between the genuine user and the imposters: 25 instances of genuine data and 225 instances of imposter data. If the data on both subjects are equal, for example, 100 instances of genuine data and 100 instances of imposter data, then the accuracy on both confusion matrix and ROC curve will be the same.

In Tables 6 and 7, the results on gesture “205” shows higher performance than those of gesture “7631.” It is because the proposed algorithm adopts smaller acceptance threshold for the training data of gesture “205” than that for gesture “7631” in the experiment. Hence, the accuracy results for gesture “7631” with larger acceptance threshold turn out to be lower than those for gesture “205.” The reason why we

TABLE 7: The summarized results of both gesture “205” and gesture “7631” using ROC curve. Low EER and high accuracy indicate higher performances. “Average AR” denotes the AR method using the average acceptance rate of all fingers and “Each AR” means the AR method using the acceptance rate of each finger.

Gesture	Evaluation	EER	Accuracy
205	ROC curve (M&SD)	0.1130	0.8870
	ROC curve (Average AR)	0.1000	0.9000
	ROC curve (Each AR)	0.1000	0.9000
7631	ROC curve (M&SD)	0.1875	0.8125
	ROC curve (Average AR)	0.2000	0.8000
	ROC curve (Each AR)	0.1958	0.8042

TABLE 8: Comparison of different biometrics approaches.

Type of biometrics	Keystroke dynamics	Mouse biometrics	Touch biometrics	Signature verification	Gesture
Naturalness	Unnatural	Unnatural	Seminatural	Seminatural	Natural
Versatility	Semiversatile (device dependent)	Not versatile	Semiversatile (device dependent)	Semiversatile (device dependent)	Versatile
Ease of use	Not easy	Not easy	Moderate	Moderate	Easy
Degree of freedom	Low	Low	Low	High	High
Error in recognition	Low	Low	Low	Moderate	High
Cost	Low	Low	Moderate	Moderate	High

adopt large acceptance threshold for gesture “7631” is due to the complications and difficulties in forming gesture “7631” as shown in Figure 7 and discussed in Section 4.1.

Table 8 shows the comparison of different biometrics approaches. As shown in the table, gesture-based biometrics has overall more advantages over other biometrics approaches. First of all, gesture-based biometrics is more natural for human to machine interaction. It is also more versatile than other approaches, because the device for gesture recognition can be implemented as small as a microchip [5]. Finally, it is easy to use, since humans usually communicate with other people using gesture, and there can be many distinct gesture patterns available.

From those systematic experiments, we have found that gesture-based biometrics has a few drawbacks in recognizing gesture such as low precision and expensive cost. We expect that as technology advances, these precision and price issues will diminish to widen a way for more gesture related applications.

As for gesture recognition, there are a few different types of gestures that can be used by humans, such as body gesture, head gesture, and hand gesture. We would like to note on the classifications of gesture because those different types of gesture and corresponding solutions often cause confusion to the readers. Table 9 shows the different type of gesture recognition that are available and their advantages and disadvantages.

Note that head gesture is not included in the table. For gesture-based control or user authentication, head gesture is impractical because many people will get nauseous while moving their heads around.

As can be seen from Table 9, stationary hand gesture has definitely more advantages over other gesture recognition in applications which need a lot of gestures. We can allegorically consider that stationary hand gesture is a motion picture while static gesture is a static image. In terms of versatility, stationary hand gesture only needs small device for recognition, whereas body gesture [35] and gait [36, 37] require larger device, for example, Leap Motion controller for hand gesture and Microsoft Kinect for body gesture. That is, the gesture space for stationary hand gesture is smaller compared to body gesture and gait which need more than a normal person space to perform. Stationary hand gesture is much more difficult to copy or mimic as it can easily be concealed from other people by just covering it with a box or your body. Finger gesture on the other hand may be difficult to conceal as it sometimes needs vision cue for the user to prevent misplacing in drawing or writing. Chances of stationary hand gesture to be implemented into other devices seem higher than other gestures as it is much more versatile and needs less space to perform. In summary, although stationary hand gesture-based user authentication has a moderate difficulty in performing because stationary hand gesture can generate more gesture patterns, this difficulty can secure the authentication from replay attack by imitation.

6. Conclusion and Future Work

Our approach, ED-FPDI in this paper has demonstrated a way to authenticate user using hand gesture with a significantly high accuracy of over 0.8 rate. This experiment is, of course, conducted under the assumption that the imposters

TABLE 9: Comparison among different gesture recognition.

Gesture type	Static gesture	Body gesture	Gait	Arm gesture	Finger gesture	Stationary hand gesture
Recognition	Static image	Motion	Motion	Motion	Motion	Motion
Versatility of device	Dependent on gesture	Not versatile	Not versatile	Not versatile	Versatile	Versatile
Gesture space	Dependent on gesture	Big	Big	Moderate	Moderate	Small
Difficulty in performing	Easy	Difficult	Easy	Easy	Moderate	Moderate
Ease of mimicking (due to difficulty of concealing gesture)	Very easy	Easy	Easy	Easy	Moderate	Difficult
Chances of implementation	Impractical but possible	Low	Low	Moderate	High	High

have already known the user's password gesture. It is worthwhile noting that, in an authentication system, false acceptance is very serious, so it has to be as low as possible, and our proposed algorithm has shown that the EER is as low as 0.2. It may have shown higher performance, if

- (1) all fingers are taken into account;
- (2) there is no hardware limitation or inaccuracy.

Hand gesture as password or authentication may not be used frequently now, but, with the upcoming technology, "smart" user interface will be implemented into most electronic devices, home appliances, vehicles, and other applications where the interfaces are mostly using gesture recognition as an input method. Our experiments indicate that hand gesture authentication shows promising future research opportunity. This is just the beginning of hand gesture authentication; therefore, more research and work have to be done before it can be used for critical authentication [11].

For our future work, we plan to record more hand gesture datasets into the experiment. In addition, we will explore new approaches to take the thumb and the little finger into account which may increase the performance. Finally, we will consider detailed analysis and improvement of Leap Motion controller as one of our future research directions.

Competing Interests

The authors declare that they have no competing interests.

References

- [1] R.-D. Vatavu and I.-A. Zaiti, "Leap gestures for TV: insights from an elicitation study," in *Proceedings of the ACM International Conference on Interactive Experiences for TV and Online Video (TVX '14)*, pp. 131–138, ACM, 2014.
- [2] M.-J. Lee, "Switch off living room light with you tv?", 2014, <http://blogs.wsj.com/digits/2014/03/10/switch-off-living-room-lights-with-ya>.
- [3] J. Song, G. Sörös, F. Pece et al., "In-air gestures around unmodified mobile devices," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology (UIST '14)*, pp. 319–329, ACM, Honolulu, Hawaii, USA, October 2014.
- [4] K. Russell, "The aria lets you control your smart watch without touching it," 2015, <https://techcrunch.com/2015/05/22/the-aria-lets-you-control-your-smart-watch-without-touching-it/>.
- [5] GoogleATAP, Project soli, <https://www.google.com/atap/project-soli/>.
- [6] C. Rogers, "Car dashboard controls just a hand gesture away," 2015, <http://www.wsj.com/articles/car-dashboard-controls-just-a-hand-gesture-away>.
- [7] P. Daugherty, O. Schybergson, and H. J. Wilson, "Gestures will be the next interface for the internet of things," 2015, <https://hbr.org/2015/07/gestures-will-be-the-interface-for-the-internet-of-things>.
- [8] S. Fong, Y. Zhuang, I. Fister, and I. Fister Jr., "A biometric authentication model using hand gesture images," *BioMedical Engineering Online*, vol. 12, article 111, pp. 1–18, 2013.
- [9] M. A. M. Shukran and M. S. B. Ariffin, "Kinect-based gesture password recognition," *Australian Journal of Basic and Applied Sciences*, vol. 6, no. 8, pp. 492–499, 2012.
- [10] G. Ceste, A. D. Giorgio, D. Molin, and J. Turesson, "Gesture-based password recognition," Project Report, 2013.
- [11] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uWave: accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657–675, 2009.
- [12] K. Lai, J. Konrad, and P. Ishwar, "Towards gesture-based user authentication," in *Proceedings of the IEEE 9th International Conference on Advanced Video and Signal-Based Surveillance (AVSS '12)*, pp. 282–287, September 2012.
- [13] J. Wu, J. Konrad, and P. Ishwar, "Dynamic time warping for gesture-based user identification and authentication with Kinect," in *Proceedings of the 2013 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '13)*, pp. 2371–2375, IEEE, Vancouver, Canada, May 2013.
- [14] F. Okumura, A. Kubota, Y. Hatori, K. Matsuo, M. Hashimoto, and A. Koike, "A study on biometric authentication based on arm sweep action with acceleration sensor," in *Proceedings of the International Symposium on Intelligent Signal Processing and Communications (ISPACS '06)*, pp. 219–222, IEEE, Yonago, Japan, December 2006.

- [15] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4–20, 2004.
- [16] B. Sayed, I. Traore, I. Woungang, and M. S. Obaidat, "Biometric authentication using mouse gesture dynamics," *IEEE Systems Journal*, vol. 7, no. 2, pp. 262–274, 2013.
- [17] J. Ho and D.-K. Kang, "Sequence alignment of dynamic intervals for keystroke dynamics based user authentication," in *Proceedings of the IEEE 15th International Symposium on Soft Computing and Intelligent Systems (SCIS), Joint 7th International Conference on and Advanced Intelligent Systems (ISIS '14)*, pp. 1433–1438, Kita-Kyushu, Japan, December 2014.
- [18] E. Farella, S. O'Modhrain, L. Benini, and B. Riccò, "Gesture signature for ambient intelligence applications: a feasibility study," in *Pervasive Computing*, pp. 288–304, Springer, New York, NY, USA, 2006.
- [19] N. Sae-Bae, K. Ahmed, K. Isbister, and N. Memon, "Biometric-rich gestures: a novel approach to authentication on multi-touch devices," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 977–986, ACM, May 2012.
- [20] A. Chan, T. Halevi, and N. Memon, "Leap motion controller for authentication via hand geometry and gestures," in *Human Aspects of Information Security, Privacy, and Trust*, T. Tryfonas and I. Askoxylakis, Eds., vol. 9190 of *Lecture Notes in Computer Science*, pp. 13–22, Springer, Berlin, Germany, 2015.
- [21] H. Liu, D. Guo, and F. Sun, "Object recognition using tactile measurements: kernel sparse coding methods," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 656–665, 2016.
- [22] H. Liu, Y. Yu, F. Sun, and J. Gu, "Visual-tactile fusion for object recognition," *IEEE Transactions on Automation Science and Engineering*, 2016.
- [23] Y. Fang, K. Wang, J. Cheng, and H. Lu, "A real-time hand gesture recognition method," in *Proceedings of the IEEE International Conference on Multimedia and Expo*, pp. 995–998, Beijing, China, July 2007.
- [24] J. Li, L. Zheng, Y. Chen, Y. Zhang, and P. Lu, "A real time hand gesture recognition system based on the prior facial knowledge and SVM," *Journal of Convergence Information Technology*, vol. 8, no. 11, pp. 185–193, 2013.
- [25] N. Aimaiti and X. Yan, *Gestire-based interaction and implication for the future [M.S. thesis]*, Umea University, Umeå, Sweden, 2011.
- [26] R. Lockton, "Hand gesture recognition using computer vision," 4th Year Project Report, 2002.
- [27] BehavioSec, "The role of biometrics in it security and continuous authentication," <http://behaviosec.com/wp-content/uploads/2012/01/The-Role-of-Biometrics-in-accessed>.
- [28] F. Weichert, D. Bachmann, B. Rudak, and D. Fisseler, "Analysis of the accuracy and robustness of the leap motion controller," *Sensors*, vol. 13, no. 5, pp. 6380–6393, 2013.
- [29] J. Guna, G. Jakus, M. Pogačnik, S. Tomažič, and J. Sodnik, "An analysis of the precision and reliability of the leap motion sensor and its suitability for static and dynamic tracking," *Sensors*, vol. 14, no. 2, pp. 3702–3720, 2014.
- [30] R. Balowin, "Why the leap is the best gesture-control system we've ever tested," 2012, <https://www.wired.com/2012/05/why-the-leap-is-the-best-gesture-control-system-weve-ever-tested/>.
- [31] J. Han and N. Gold, "Lessons learned in exploring the leap motion tm sensor for gesture-based instrument design," in *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 371–374, London, UK, June 2014.
- [32] W. J. Masek and M. S. Paterson, "A faster algorithm computing string edit distances," *Journal of Computer and System Sciences*, vol. 20, no. 1, pp. 18–31, 1980.
- [33] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874, 2006.
- [34] J.-M. Cheng and H.-C. Wang, "A method of estimating the equal error rate for automatic speaker verification," in *Proceedings of the 2004 4th International Symposium on Chinese Spoken Language Processing (ISCSLP '04)*, pp. 285–288, IEEE, Hong Kong, December 2004.
- [35] B.-W. Hwang, S. Kim, and S.-W. Lee, "A full-body gesture database for automatic gesture recognition," in *Proceedings of the 7th International Conference on Automatic Face and Gesture Recognition (FGR '06)*, pp. 243–248, IEEE, April 2006.
- [36] L. Wang, T. Tan, H. Ning, and W. Hu, "Silhouette analysis-based gait recognition for human identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1505–1518, 2003.
- [37] A. Muro-de-la-Herran, B. García-Zapirain, and A. Méndez-Zorrilla, "Gait analysis methods: an overview of wearable and non-wearable systems, highlighting clinical applications," *Sensors*, vol. 14, no. 2, pp. 3362–3394, 2014.

